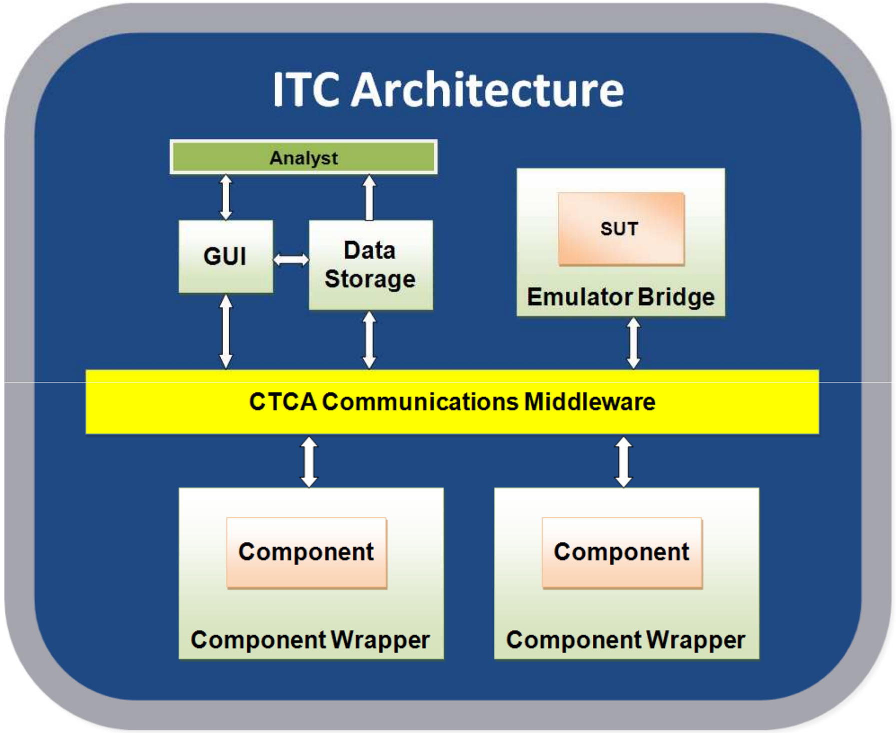
ITC

- To provide the capability to fully verify and validate software systems, the IV&V Facility needs a mechanism to perform dynamic analysis across multiple NASA missions. As a result, the Independent Test Capability (ITC) Research and Development team is developing a middleware-based solution that provides a common graphical user interface and APIs for simulation development, maintenance, and operation. This presentation presents the simulation architecture, tools, and environment followed by a demonstration.

ITC
Independent Test Capability



Independent Test Capability

September 16, 2010

Presenters:
Justin Morris,
Steven Seeger

Team Members
Justin Morris, ITC Lead
Steven Seeger, Lead Software Engineer
Brandon Bailey
Shawn Carroll
Jeff Joltes
Justin McCarty
Dan Nawrocki
Peter Thompson
Mike Wise
ivv-itc@lists.nasa.gov

Agenda

- Background
- Objectives
- ITC Architecture
- ITC Demonstration
- Summary

IV&V Current Capability

- Verify and Validate Implementation (WBS 6.0)
  - Currently is performed via static analysis only

	Code Analysis (Manual and Auto)	Test Result Analysis
Process	1. IV&V receives Build N from Project 2. Compare differences with previous builds 3. (manual)Perform line-by-line walkthrough of code 4. (auto) Static analysis tools exercised against codebase and examine output results	1. IV&V receives test results from Project 2. Examine actual test results against expected test results 3. Identify any discrepancies 4. Identify independent test scenarios
Value	Provides an additional set of eyes on software components Can be applied early in the lifecycle (auto) Provides set of rules to execute against code base to identify critical bugs such as memory leaks, buffer overflows, and NULL pointer dereferences	Confirms intended system behavior Provides input into test scenario development

IV&V Missing Capability

1. **Provides mechanism to complement current testing**
   - Identify run-time vulnerabilities, test system interfaces, execute desirable software tests that have been de-scoped by the Project
   - Without this capability, IV&V relies on tests developed by the Project

2. **Provides ability to automatically verify sets of requirements**

3. **Provides improved IV&V results**
   - Increases IV&V capabilities and analysts' understanding of system dynamics
   - Validates static analysis findings and identifies false findings from static analysis techniques
   - Validate issues (TIMS) found from other IV&V activities

Static versus Dynamic

Static Analysis		Dynamic Analysis	
Advantages	**Disadvantages**	**Advantages**	**Disadvantages**
• Finds weaknesses in exact location • Allows quicker turnaround for fixes • Finds errors earlier in lifecycle *Automated Tools* • Relatively fast • Can scan all of code	• Can be time consuming • False Positives and False Negatives • Requires trained personnel • Do not provide runtime vulnerabilities *Automated Tools* • Can provide false sense of security • Only as good as rules they are using	• Finds run-time vulnerabilities • Provide increased flexibility of what to look for • Identifies vulnerabilities that may have been false negatives in static analyses • Validation of Static Analysis Findings	• Can provide false sense of security that everything has been addressed • False Positives and False Negatives • Requires trained personnel

Independent Test Capability

- **Internal Research and Development Team**
  - R&D advances processes, tools, and knowledge through the exploration and integration of practical solutions.

- **Charter**
  - Develop, maintain, and operate an adaptable test environment for the IV&V program that enables the dynamic analysis of software behaviors for multiple NASA missions.
    - ITC = experts in simulation
    - IV&V Project = experts in systems
  - Provide a means to assess system dynamics against IV&V's Three Questions
    - Test what the system does
    - Test what the system is not supposed to do
    - Test what the system does under adverse conditions

Objectives

R&D advances processes, tools, and knowledge through the exploration and integration of practical solutions.

Tools

Solutions

Knowledge

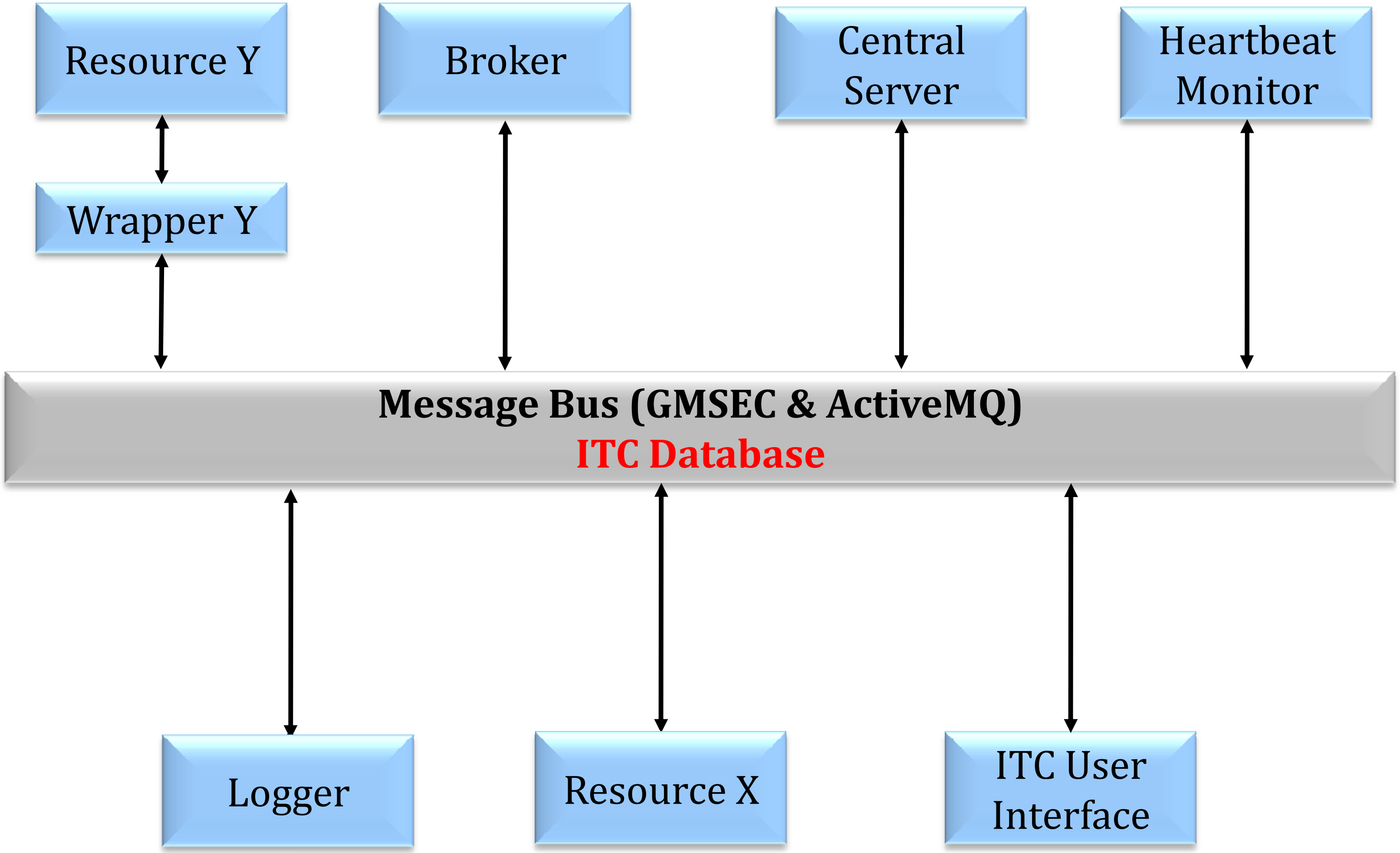
	Objectives	Means to Achieve
1	Develop framework to support the integration of software simulation and testing tools	Developing ITC toolset that provides common Graphical User Interface to support software simulation and testing tools (Demo)
2	Perform dynamic analysis on a current mission	Collaborating with GSFC GO-SIM effort to build software-only simulator (Demo)
3	Infuse ITC approach into IV&V current processes and tasks	Process: ITC SLP Products: Independent Test Plan and Test Report

ITC Framework Goals and Objectives

- Provide a simulation and test environment that enables the dynamic analysis of software through the integration of simulation tools, models and test articles
- A software test environment that is easily adaptable and reconfigurable for multiple projects to be used facility-wide
- Stimulate the developers' software running on the intended hardware or emulated hardware
- APIs for use by development projects
- Common User Interface

Independent Test Capability

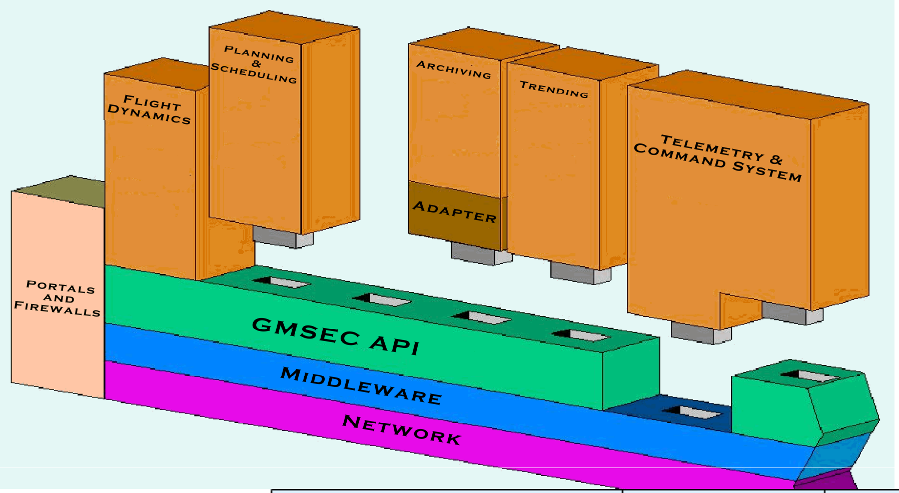
ITC Architecture



GMSEC

Goddard Space Flight Center (GSFC) Mission Services Evolution Center

By creating a "framework", individual applications can be easily integrated into an existing system without regard to many underlying implementation details.

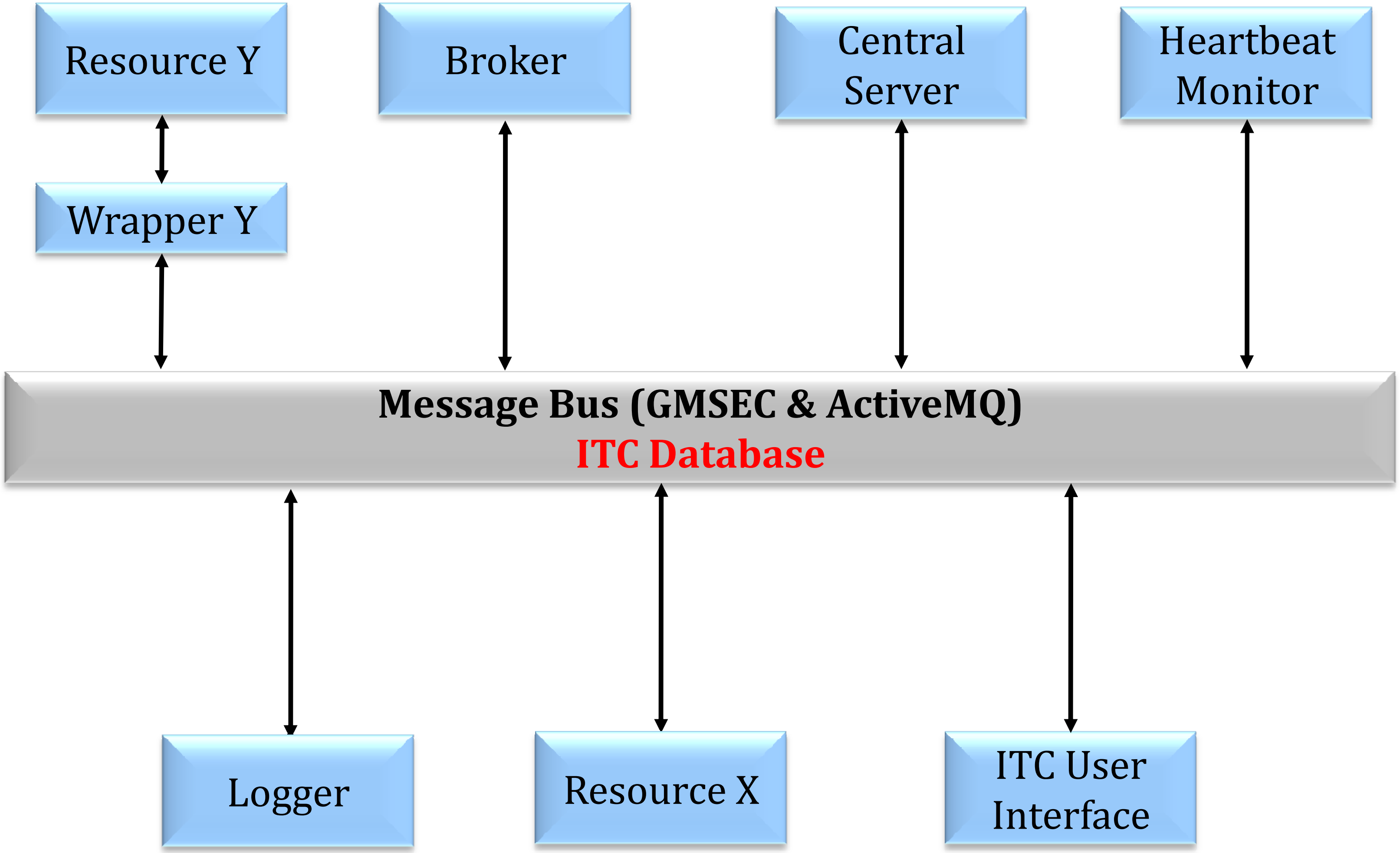


Components	Telemetry & Command	Automation	Flight Dynamics		
	Planning	Monitoring	Archive & Assessment	Simulators	
GMSEC Messages	Telemetry Frame	Log	Directive Request	Directive Reply	
	Scheduling	Mnemonic Value	Comp. to Comp. Transfer		
GMSEC API	GMSEC Applications Programming Interface C, C++, Java, Perl, *Python, Delphi*				
Middleware	Rendezvous	TIBCO SmartSockets	GSFC Bus	ICE	SCL SWB
Operating Systems	Windows	Linux	HP-UX	Solaris	Mac OS X

Slide referenced from GMSEC Presentation "Getting Started Programming on GMSEC Information Bus"

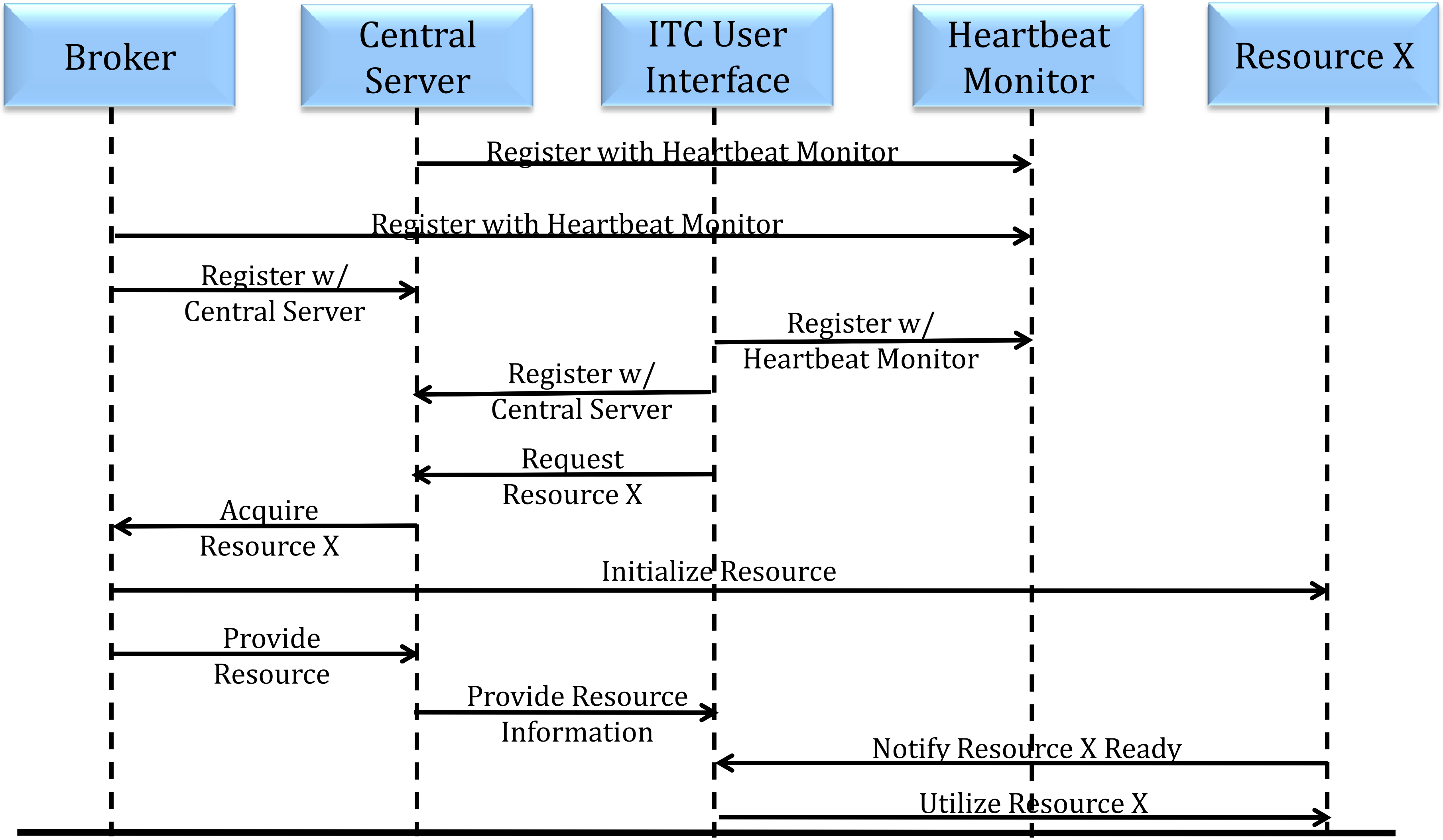
ITC Independent Test Capability

ITC Architecture



Independent Test Capability

ITC Nominal Operation



ITC

Independent Test Capability

ITC DEMONSTRATION

ITC Summary

- Reusable Simulation Components
- Common Interface across missions
- Integrate simulation tools across projects
- Reduction in training to setup and configure operational simulator
- Potential to provide simulator to many IV&V analysts with no risk of “breaking” hardware
- No hardware maintenance

Conclusion

- Background
- Objectives
- ITC Architecture
- ITC Demonstration
- Summary

QUESTIONS?